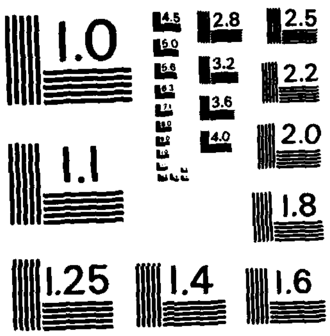
AD-A160 128 THE ROLE OF INTELLIGENT REACTIVE PROCESSING IN PRODUCTION MANAGEMENT(U) CARNEGIE-MELLON UNIV PITTSBURGH PA ROBOTICS INST M S FOX ET AL. 04 APR 85 1/1

UNCLASSIFIED AFOSR-TR-85-0761 F49620-82-K-0017 F/G 13/8 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A160 128

(2)

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY ---	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S) AFOSR-TR- 85-0761
6a. NAME OF PERFORMING ORGANIZATION Carnegie-Mellon University / 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION AFOSR
6c. ADDRESS (City, State and ZIP Code) The Robitics Institute Schenley Park Pittsburgh, PA 15213	7b. ADDRESS (City, State and ZIP Code) Bldg. 410 Bolling AFB, D.C. 20332-6448
8a. NAME OF FUNDING/SPONSORING ORGANIZATION AFOSR / 8b. OFFICE SYMBOL (If applicable) NM	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F49620-82-K-0017
8c. ADDRESS (City, State and ZIP Code) Bldg. 410 Bolling AFB, D.C. 20332-6448	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO. 61102F; PROJECT NO. 2304; TASK NO. A7; WORK UNIT NO.

11. TITLE (Include Security Classification)
The Role of Intelligent Reactive Processing in Production Management

12. PERSONAL AUTHOR(S)
Mark S. Fox and Stephen F. Smith

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
Technical	FROM ____ TO ____	4 April 1985	3

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	Intelligent systems, production management
XXXX	XXXXXXXXX	XXX	

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Our work has been concerned with the construction of intelligent systems for production management and control. This paper focuses on the reactive capabilities that such systems must possess to be of practical use in dynamic environments. These capabilities include omonitoring events on the factory floor, identifying deviations from predicted production schedules, and intelligent schedule repair.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☒ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL Dr. Robert N. Buchal	22b. TELEPHONE NUMBER (Include Area Code) (202)767-4939 / 22c. OFFICE SYMBOL NM

DD FORM 1473, 83 APR — EDITION OF 1 JAN 73 IS OBSOLETE.

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

AFOSR-TR- 85 - 0761

THE ROLE OF INTELLIGENT REACTIVE PROCESSING IN PRODUCTION MANAGEMENT[1]

by

Mark S. Fox and Stephen F. Smith
The Robotics Institute, Carnegie-Mellon University
USA

ABSTRACT

Our work has been concerned with the construction of intelligent systems for production management and control. This paper focuses on the reactive capabilities that such systems must possess to be of practical use in dynamic environments. These capabilities include monitoring events on the factory floor, identifying deviations from predicted production schedules, and intelligent schedule repair.

1. Introduction

Manufacturing in a job shop environment is composed of activities that can and must be managed at different levels of abstraction. A shop floor can be viewed as a group of work areas; a work area is composed of manufacturing cells; and a manufacturing cell is composed of individual machines, robots, and tools. There are two distinct aspects to production management in such environments. The first concerns an ability to effectively *predict* shop behavior through the generation of production plans. Appropriate operations must be selected, and resources must be assigned and scheduled at each level of abstraction. Job shop scheduling is a complex activity that is influenced by knowledge accumulated from many different sources in the plant, and automation of this function requires an effective strategy for utilizing this knowledge in the development of schedules. However, an ability to generate realistic production schedules only addresses half of the problem of production management. There is a second aspect that concerns an ability to *react* to changing circumstances. The shop floor is a dynamic environment where unexpected events continually occur and quickly force changes to planned activities. Hence, the automation of decision making for production management must involve not only the prediction of shop behavior through planning, but also the ongoing alteration of plans in reaction to unexpected events.

In this paper we focus on the reactive capabilities that a production management system must possess to be of practical use in a dynamic job shop environment. We explore the major issues involved, considering, in turn

- the monitoring of events on the factory floor,
- the identification of deviations from predicted production plans,
- the alteration or repair of invalidated production plans, and

[1]This research was supported, in part, by the Air Force Office of Scientific Research under contract F49620-82-K0017, Westinghouse Electric Corporation, and the Robotics Institute

(Appeared in: Proceedings of CAM-I's 13th Annual Meeting and Technical Conference, Clearwater Beach, Florida, November 13-15, 1984)

85 10 11 161

- the improvement of subsequent predictions through analysis of detected deviations.

As is the case with the predictive planning function, we argue that intelligent reactive processing is a knowledge intensive activity and suggest a knowledge-based approach to providing such a capability. To provide a framework for the discussion, we begin by outlining our approach to the representation and generation of production plans.

2. Representation and Generation of Production Plans

Analysis of the job shop scheduling domain has indicated that the crux of the scheduling problem is the determination and satisfaction of a large variety of constraints. Schedules are influenced by such diverse and conflicting factors as due date requirements, cost restrictions, production levels, machine capabilities and substitutability, alternative production processes, order characteristics, resource requirements, and resource availability. In adopting a knowledge-based approach to job shop scheduling, we have sought to explicitly represent *all* relevant scheduling knowledge as constraints in the system's knowledge base, and to cast schedule construction as a *constraint-directed* heuristic search that is driven by this knowledge. The result is a general methodology for scheduling that allows the incorporation of all constraints deemed relevant by the user. Our work with the ISIS job shop scheduling system [1, 2, 4] has demonstrated the viability of this approach.

In representing a given constraint, it is necessary to capture the full range of information about the constraint that is necessary in constructing satisfactory schedules. Since constraints are often conflicting in nature, a central representational concern is that of *relaxation*. Accordingly, the specification of allowable alternatives, expressed either in the form of predicates or choice sets, is a prominent feature of the constraint representation. The association of a *utility* with each relaxation specified in a constraint provides a means of designating preferences amongst the alternatives available, intuitively indicating the degree to which the constraint is satisfied if the associated relaxation is chosen. Other salient features of the constraint representation include the *importance* of satisfying the constraint, the constraint's *relevance* to the scheduling decisions that have to be made, and the constraint's *interdependencies* with other constraints.

The packaging of all relevant scheduling knowledge as constraints in the knowledge base enables the use of a fairly general search procedure as a means of generating production schedules. Within this *constraint-directed reasoning* approach, constraints are used both to bound the generation of possible solutions and to focus selection amongst the alternatives generated. For example, the "next-operation" of a given operation is viewed as a precedence constraint and the due date for an order as a goal constraint. Constraints of the former variety can be used to elaborate the solution space of partial schedules during the search while the latter is used to rate schedules in that space. Constraint knowledge is also used to detect and diagnose poor solutions produced by the search. The utilities assigned by the constraints in rating the solution provide a basis for detecting poorly satisfied constraints, and the interdependencies amongst constraints provide guidance in identifying the cause of the problem (e.g. a poor decision with respect to a related constraint). The production schedules that are generated serve as additional constraints for any subsequent scheduling that must be performed. The reader is referred to [4] for a more detailed discussion of constraint-directed reasoning.

Much of the constraint knowledge utilized in the generation of production schedules is also relevant in the context of reactive processing and, as such, it is felt that a constraint-based paradigm offers a fruitful approach to providing this functionality. The remainder of this paper explores the types of constraint knowledge required to support intelligent reactive processing.

3. Monitoring

An ability to monitor ongoing work on the shop floor and detect unexpected events is fundamental to a reactive capability. We can identify two distinct levels of monitoring that are required, each with distinguishable characteristics and outcomes.

The first level at which monitoring must take place is at the *process* level. Here, the monitoring is concerned with detecting problems in low level manufacturing processes (e.g. a machine malfunction), and is driven by sensors and other automatic information gathering devices. Data is sampled continuously in real time, and the monitoring process must be robust in the face of spurious readings and sensor malfunctions.[2] Upon detection of a problem at this level, the course of action to take is typically clear cut (e.g. shut the machine down and call the operator). However, once the action is taken, an inconsistency is introduced between predicted and actual shop behavior. Thus, the actions taken at this level constitute unexpected events that must be detected and reacted to at a higher level (see below).

The second level at which monitoring must take place, and the level with which we are most concerned in this paper, is the *production activity* level. Sampling at this level is event-based, and is driven by manual input from various system interfaces as well as messages received from lower level control processes. Input can range from simple status updates such as an indication that a particular operation has completed or that a particular machine has gone down to more far reaching events such as a change in production goals. The task here is to monitor the incoming updates to the shop model and and detect situations where the actual shop behavior deviates from system predictions. If we adopt a constraint-based view, this amounts to a comparison between the predicted constraints in the model (e.g. the resource reservations contained in the production schedule) and the constraints which have resulted from plant operation. In some cases, the comparison is simply a test for conflicts (e.g. a machine down time constraint might conflict with the machine reservations of orders), while in other cases the comparison might entail an evaluation of the predicted constraints with respect to the newly imposed constraints (e.g. if a change in a production goal occurs, and an order's schedule is judged to satisfy this constraint very poorly, then rescheduling might be warranted).In either case, the constraint-based perspective offers a direct approach to the identification of deviations.

4. Repair

While the detection of deviant behavior appears to be rather straightforward, determination of the effects of the deviation, and consequently the appropriate repair action to take, can be quite difficult. For example, the repair action required for a machine malfunction might be localized to a particular work area (e.g. simply rerouting affected orders through alternative machines), while a change in production goals may require a complete rescheduling of the plant. There might also exist alternative repairs with respect to a given deviation. In this case additional knowledge, such as the urgency of the repair, estimates of the computational effort associated with carrying out each alternative, and the system's belief in the certainty of alternative repairs must enter into the decision as to how to proceed.

One approach to determining the effect of change focuses on goal-directed rule-based processing [5]. For each category of error (or, in our terms, each type of constraint that may result from the detection of an unexpected event), an appropriate repair procedure to follow is specified. Such an approach is appropriate in situations where knowledge of the effect of events is fairly complete. However, as is the case with all rule-based systems,

---

[2] See [3] for a more detailed discussion of these issues and an approach to dealing with them.

problems lying outside of the scope of the rule set may not be acted upon properly. Nonetheless, such an approach can provide a useful framework for reacting to well understood deviations, with the system falling back on more general reasoning strategies when the rule set does not apply.

Knowledge of constraint interdependencies, which define the extent to which the selection of values for individual constraints effect (or constrain) the selection of values for other constraints, can provide the basis for a more general approach to schedule repair. Consider the case where the poor satisfaction of the due date constraints associated with scheduled orders is detected. Clearly, the ability to satisfy the due date constraint is dependent on the number of shifts that various work areas in the shop are operating, and the appropriate repair action, in this case, may be to increase the number of shifts. More generally, a constraint may have interdependencies with several other constraints, suggesting alternative directions along which the repair action might proceed. In such cases, knowledge of the sensitivity of the individual interdependencies involved, as well as the level of abstraction at which the related constraints reside (as defined by their positions in the overall network of interdependencies) can provide a means for determining which direction to take. Once a specific constraint has been identified as the cause of the deviation, repair action can be effected in different ways. A specific action might be inherited via the constraint taxonomy that structures the various constraint types known to the system, providing the capabilities of the rule-based approach discussed above. Alternatively, the interdependency network may be associated with levels in a hierarchical system, in which case there is a direct mapping between the constraint causing the deviation and the particular level of processing required.

A goal of any effort to repair predicted plans that have become invalidated is that of minimizing the extent of the change. Shop stability is an important concern and we would like the revised schedule to deviate as little as possible from previous schedules. Toward this end, the ISIS scheduling system illustrates the advantages of a constraint-directed reasoning approach in its approach to rescheduling an order that has had resource reservations bumped. ISIS transforms the order's reservations into preference constraints that focus the rescheduling effort toward prior solutions if they remain feasible. Only if the prior solution is now infeasible will the reservations be discarded.

5. Learning: the transition from reaction to prediction

A larger issue than that of intelligently reacting to unexpected events concerns providing the system with an ability to improve its predictions on the basis of the events it has encountered in the past. Recurring deviations may be symptomatic of an inaccurate or incomplete model of the specific job shop environment, and the system should take steps to rectify the misconception. For example, if a given machine is continuously breaking down, this knowledge should be taken into account during the planning process.

The processes involved involved transforming reactive experience into knowledge that can be applied to improve the system's predictive ability may not be that unlike those that have been described above, except that they are operating at a meta-level. In the simplest case, a well defined class of recurring events are defined. Monitoring processes operate on a recorded history of the unexpected events that have been encountered, and a set of rules map specific recurring events to the addition of specific constraints to the shop model.

6. Conclusion

In this paper we have attempted to lay out the issues involved in providing an ability to intelligently react to dynamic changes in the state of the shop floor. In doing so we have advocated a constraint-based approach and have identified the types of constraint knowledge that appear relevant to providing an intelligent reactive processing capability.

References

1. Fox, M.S., Smith, S.F., Allen, B.P., Strohm, G.A., and Wimberly, F.C. ISIS: A Constraint-Directed Reasoning Approach to Job Shop Scheduling. Proceedings IEEE Conference on Trends and Applications 83, Gaithersburg, MD, May, 1983.

2. Fox, M.S. *Constraint-Directed Search: A Case Study of Job Shop Scheduling*. Ph.D. Th., Carnegie-Mellon University, 1983.

3. Fox, M.S., Lowenfield, S., and Klienosky, P. Techniques for Sensor-Based Diagnosis. Proceedings 8th International Joint Conference on Artificial Intelligence, Karlsruhe, West Germany. August, 1983.

4. Fox, M.S. and Smith, S.F. "ISIS: A knowledge-Based System for Factory Scheduling." *Expert Systems 1*, 1 (July 1984), 25-49.

5. Haley, P., and McDermott, J. PTRANS: A Rule-Based Management Assistant. Computer Science Department, Carnegie-Mellon University, 1984.

END

FILMED

11-85

DTIC